



## Visão

## 1. Histórico de Revisões

| Nome         | Função |
|--------------|--------|
| Breno Campos | Full   |
| -            |        |

| Data       | Versão | Descrição  | Autor        |
|------------|--------|--|--------------|
| 14/09/2019 | 1.0    | Criação e desenvolvimento da ideia e da documentação. Definição e resolução do problema, análise e descrição dos requisitos. | Breno Campos |
|            |        |  |              |

## 2. Ambiente do usuário

## 3. Declaração

|                                  |   |
|----------------------------------|---|
| O problema de                    | Entregar produtos e serviços em residências.  |
| Afeta                            | Donos de estabelecimentos (alimentação) e clientes.   |
| O impacto do qual é              | Muitos estabelecimentos ainda não se modernizaram perante a um mundo cada vez mais digital e conectado. Há casos de empresas que agendam pedidos no "papelzinho" e se comunicam com clientes via WhatsApp, podendo tornar o trabalho mais confuso e desorganizado, para ambos os lados. |
| Uma solução bem-sucedida deveria | Um aplicativo para facilitar a comunicação e o agendamento entre a empresa e os clientes.   |

## 4. Declaração do Produto

|             |   |
|-------------|---|
| Para        | Estabelecimentos comerciais.  |
| Que         | Entregam produtos na residência de clientes.  |
| O YeahFood! | É um aplicativo de delivery.  |
| Que         | Permite que clientes possam realizar pedidos e a empresa possa atender esses pedidos. |

|               |  |
|---------------|--|
| A menos que   |  |
| Nosso produto |  |

## 5. Resumo dos Interessados

| Nome          | Descrição  |
|---------------|--|
| Cliente       | É o ator que fará pedidos à empresa                                    |
| Administrador | É o ator responsável por gerenciar e realizar os pedidos aos clientes. |

## 6. Premissas e Dependências

Servidor e hospedagem

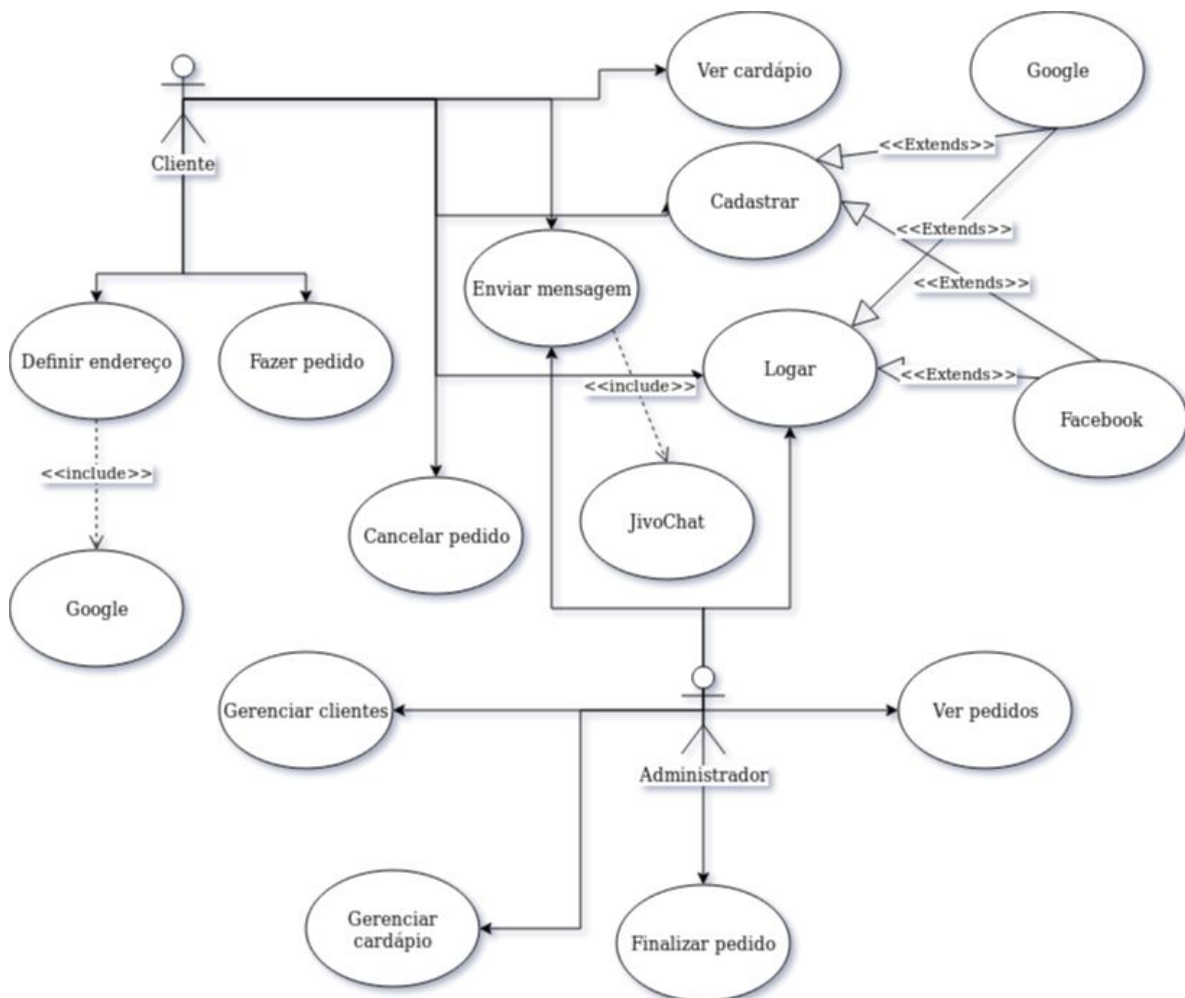
## 7. Requisitos Funcionais e Recursos

| Código | Requisitos   | Prioridade |
|--------|--|------------|
| RF01   | 0 cliente pode criar uma conta.                                | Alta       |
| RF02   | 0 cliente pode logar.  | Alta       |
| RF03   | 0 cliente pode realizar pedidos.                               | Alta       |
| RF04   | 0 cliente pode cancelar seu pedido.                            | Baixa      |
| RF05   | 0 cliente pode ver o cardápio.                                 | Alta       |
| RF06   | 0 cliente pode definir o endereço (localização).               | Alta       |
| RF07   | 0 cliente pode se comunicar com o administrador em tempo real. | Baixa      |
| RF08   | 0 administrador pode logar.                                    | Alta       |
| RF09   | 0 administrador pode ver os pedidos pendentes.                 | Alta       |
| RF10   | 0 administrador pode finalizar pedidos.                        | Alta       |
| RF11   | 0 administrador pode gerenciar os clientes cadastrados.        | Média      |
| RF12   | 0 administrador pode gerenciar o cardápio.                     | Alta       |
| RF13   | 0 administrador pode se comunicar com o cliente em tempo real. | Baixa      |
| RF14   | 0 administrador pode gerenciar o tema do site.                 | Baixa      |

## 8. Requisitos não Funcionais

| Código | Requisito  | Prioridade |
|--------|--|------------|
| RNF01  | 0 software deve ser implementado com JavaScript. | Alta       |
| RNF02  | 0 servidor deve ser implementado com Adonis.js.  | Alta       |
| RNF03  | Os dados serão salvos em um servidor.            | Alta       |
| RNF04  | Deve ser desenvolvido para web e ser responsivo. | Alta       |
| RNF05  | Usar uma API de localização via GPS.             | Alta       |
| RNF06  | Usar uma API de notificações.                    | Média      |
| RNF07  | Usar o Jivochat para o chat.                     | Baixa      |
| RNF08  | Usar QR code para localizar a URL.               | Baixa      |
| RNF09  | Usar template bootstrap.                         | Alta       |

## 9. Diagrama de caso de uso



## **10. Equipamentos e Tecnologias Necessárias**

- a. Front-end
  - i. HTML
  - ii. JavaScript
  - iii. CSS
- b. Back-end
  - i. Adonis.js
  - ii. Node.js
  - iii. Banco de dados SQL (PostgreSQL)
  - iv. Hospedagem (Hostgator)
  - v. Chat online (JivoChat)
  - vi. Google Auth API
  - vii. Facebook Auth API
  - viii. Google Local API

## **11. Revisão da Literatura**

## **12. Referências**